REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-05-

0414

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments re; information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any ~...... penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE		3. DATES COVERED (From - To)
July 20, 2005	Final		May1, 2004 - April 30, 2005

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Rapidly Reconfigurable High Performance Computing Cluster	
	5b. GRANT NUMBER
	FA9550-04-0266
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Mark A. Richards	E-21-6RR (PeopleSoft ID 21066RR)
Daniel P. Campbell	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Georgia Institute of Technology School of Electrical & Computer Engineering 777 Atlantic Drive Atlanta, GA 30332-0250	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
Air Force Office of Scientific Research 4105 Wilson Boulevard Arlington, VA 22203-1954 NM	AFOSR/NM
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Georgia Institute of Technology (GIT) is highly active in developing middleware standards for high performance embedded computing (HPEC), especially for DoD-relevant applications in sensor signal processing and cognitive processing. To support this research, GIT has used the funding provided by this project to purchase, install, and make operational a 104-processor, 900+ gigaflops heterogeneous Beowulf computing cluster. The system is currently in use as a test bed and simulator in support of the embedded multiprocessor software programs in which GIT participates, particularly the HPEC Software Initiative (HPEC-SI) program. A detailed list of equipment purchased is provided later in this report. This system has significantly enhanced GIT's current research capabilities and allowed us to expand our contributions to HPEC programs by enabling more thorough experimentation demonstration and testing of emerging standards

15. SUBJECT TERMS

Beowulf, multiprocessor, middleware, cluster computing.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	19	Mark A. Richards
					19b. TELEPHONE NUMBER (Include area code)
					404-894-2714

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Defense University Research Instrumentation Program (DURIP):
Rapidly Reconfigurable High Performance Computing Cluster

By:

Mark A. Richards
School of Electrical and Computer Engineering

Daniel P. Campbell
Sensors and Electromagnetic Applications Laboratory
Georgia Tech Research Institute

Submitted to:
Air Force Office of Scientific Research (AFOSR)
4105 Wilson Blvd.
Arlington, VA 22203-1954
ATTN: Dr. Spencer Wu

Submitted by:

GEORGIA INSTITUTE OF TECHNOLOGY
A Unit of the University System of Georgia
Georgia Tech Research Institute
Atlanta, Georgia 30332-0800

Cage No. 1G474

Contracting through:

GEORGIA TECH RESEARCH CORPORATION
Centennial Research Building
Georgia Institute of Technology
Atlanta, Georgia 30332

July 2005

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ACRONYMS

ACIP	Architectures for Cognitive Information Processing
AFOSR	Air Force Office of Scientific Research
AI	Artificial Intelligence
API	Application Programming Interface
CCRF	Cobb County Research Facility
COTS	Commercial Off-the-Shelf
DARPA	Defense Advanced Research Projects Agency
DoD	Department of Defense
DURIP	Defense University Research Instrumentation Program
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GIT	Georgia Institute of Technology
GPU	Graphical Processing Unit
GTRI	Georgia Tech Research Institute
HPEC	High Performance Embedded Computing
HPEC-SI	High Performance Embedded Computing Software Initiative
MIT-LL	Massachusetts Institute of Technology Lincoln Laboratory
MSI	Morphware Stable Interface
PCA	Polymorphous Computing Architectures
VSIPL	Vector, Signal, and Image Processing Library

PREFACE

This document is the Final Performance Report under Georgia Institute of Technology (GIT) Project E-21-6RR, "Rapidly Reconfigurable High Performance Computing Cluster." This project is sponsored by the Air Force Office of Scientific Research (AFOSR) under the Defense University Research Instrumentation Program (DURIP) and is conducted under AFOSR grant FA9550-04-0266.

SECTION 1

EXECUTIVE SUMMARY

The Georgia Institute of Technology (GIT) is highly active in developing middleware standards for high performance embedded computing (HPEC), especially for DoD-relevant applications in sensor signal processing and cognitive processing. This activity is manifested through GIT's leadership in a number of consortia in the DoD HPEC community. These include the Vector, Signal, and Image Processing Library Forum; the High Performance Embedded Computing Software Initiative (HPEC-SI) program; the Defense Advanced Research Projects Agency's (DARPA) Polymorphous Computing Architectures program and the associated Morphware Forum; and DARPA's Architectures for Cognitive Information Processing program. GIT's role in these programs includes building community consensus, standards design and development, reference implementation, test implementation, publicity, and training.

These programs emphasize the development of software for a wide-ranging space of homogeneous and heterogeneous multiprocessors. To support this research, GIT has used the funding provided by this project to purchase, install, and make operational a 104-processor, 900+ gigaflops heterogeneous Beowulf computing cluster. The system is resident at the Georgia Tech Research Institute's Cobb County Research Facility. The system is currently in use as a test bed and simulator in support of the embedded multiprocessor software programs described above, particularly the HPEC-SI program. A detailed list of equipment purchased is provided later in this report.

This system has significantly enhanced GIT's current research capabilities and allowed us to expand our contributions to these programs by enabling more thorough experimentation demonstration and testing of emerging standards. Increased research effort in these and related programs may ultimately also augment GIT's educational capacities, by allowing expanded participation by undergraduate and graduate research assistants, as well as potentially greater student-faculty interaction through special topics courses, augmentation of existing continuing education programs, and creation of new programs.

SECTION 2

BACKGROUND AND OBJECTIVES

2.1   High Performance Computing Trends

Three important technology trends are apparent in DoD embedded high performance computing application development. The first is a trend away from extremely powerful, single-processor computing platforms toward multiprocessor systems composed of several simpler, independent, cheaper, and easier-to-maintain processors. Such multiprocessor architectures have proven to be much more efficient platforms for delivering a given amount of computational power in terms of cost, weight, power consumption, and maintainability. Nearly all modern defense computing systems now meet high performance demands through the use of multiprocessor computers.

The second trend is the increasing importance on standardization placed by the DoD, as well as the software development community at large. The process of developing *ad hoc* solutions to commonly encountered software problems is tremendously expensive and often ineffective, approach. The increasing reliance on commercial off-the-shelf (COTS) hardware introduces non-portable, platform-specific interfaces to what are actually very common, widely shared functionality. For example, competing vendors will have different function names and argument lists for calling essentially the same fast Fourier transform (FFT) subroutines. Standardization of software development approaches, frameworks, and APIs has been shown to bring several very important benefits to defense applications. Domain-specific portable APIs, rather than platform-specific APIs, have greatly improved both application development and application maintenance. Portability allows program managers to avoid committing to specific computing hardware platforms too early in the program. Deferring this decision allows application requirements to be tested and refined so that hardware decisions are based on better information, and at the same time avoids becoming locked in early to hardware that will be obsolete before a system is fielded. Portability also allows a far more efficient development cycle by allowing application developers to author and test algorithms on a different platform (for example a workstation, or cluster computer) than the target deployment platform. This often leads to faster compile and test cycles, and results in a larger set of high quality development tools available to the authors.

Standardization improves productivity in other ways. Domain APIs allow commonly encountered computing tasks to be centralized and simplified under a common naming and interface convention. These tasks then become easier to write, being both simpler and less error prone, and easier to maintain, being more rapidly understood and typically containing many fewer lines of code at the application level.

Finally, standardization has been shown to lead to increased performance. While the commonly-held belief as little as one decade ago was that hand-tuned assembly code would always produce higher performance application software, domain-specific portable APIs have been shown to produce code rivaling hand-tuned assembly performance with far less development effort. With the increased levels of complexity in modern defense applications, it is no longer possible in the vast majority of programs to hand-tune every performance-critical piece of software. Nearly all applications today are written in higher level languages, most

2

often C, C++, or Fortran. While tools, compilers, and build systems for these languages have made tremendous strides in the last twenty years, basic, non-domain-specific, general purpose languages are limited in the scope of knowledge of the application that is possible. Standardization has allowed domain-specific problems, lexicons, and algorithm descriptions to augment the general purpose languages, allowing much greater implied information flow between the application developer and the build systems. Compilers and other tools that can take advantage of this extra program information then produce code nearly as good as the best hand-tuned codes, and in some cases even better. Standardized, domain-specific APIs have been shown to allow much greater performance than *ad hoc* software written purely in general purpose languages.

The third trend is the introduction of new computational styles into DoD software, and the design of software development techniques that use knowledge of the computing domain to improve both code quality and productivity. For example, "stream computing" is a widely used paradigm that is applicable to many sensor signal processing systems and is a key component of the multiprocessor programming approach being developed by the Morphware Forum [4] (see next subsection). The newest development is the rapidly increasing interest in "cognitive computing" [5],[6]. These programs emphasize rapid access and searching of very large memories, heuristic algorithms for solving NP-complete problems, extensive feedback and adaptation, and other techniques that differ drastically from stream processing. It is not yet well understood what types of computer architectures may prove most effective in hosting cognitive applications.

2.2 Georgia Tech Activity in HPEC Software Development

The Georgia Institute of Technology (GIT) is an established leader in developing middleware standards for High Performance Embedded Computers. Since 1996, GIT has co-chaired the Vector, Signal, and Image Processing Library (VSIPL) Forum through the completion and adoption of the industry standard portable, high-performance signal processing Application Programming Interface (API) [1]. Since then, GIT has participated in the High Performance Embedded Computing Software Initiative (HPEC-SI) program, in technical and advisory roles. This program develops, prototypes, and demonstrates signal processing software standards that improve the portability, development costs, and performance of DoD embedded signal-processing applications [2]. A central aspect of this program is the study of mechanisms for optimal mapping of algorithms to multiprocessor systems. GIT also chairs the Polymorphous Computing Architectures (PCA) Morphware Forum. DARPA's PCA program develops computing platforms with high degrees of runtime configurability, usually by means of dividing processors into many sub-units, each of which may operate in several modes, with configurable high speed inter-unit communication channels [3]. In many ways, the PCA devices can be considered to be multiprocessors on a chip. The Morphware Forum, an activity of the PCA program led by GIT, seeks to establish software frameworks that allow portable, cross-platform application development, fully exploiting the performance benefits of polymorphism [4]. The DARPA Architectures for Cognitive Information Processing (ACIP) program extends addresses the development of embedded hardware and software structures for cognitive processing. Within ACIP, GIT is investigating the concept of a "Living Framework", similar in spirit, and possibly linked to, the morphware concepts of the PCA program. GIT's role in these programs variously includes building community consensus, standards design and development, reference implementation, test implementation, publicity,

and training. More recently, the U.S. Air Force Research Laboratory (AFRL) has initiated an effort to investigate the field of architectures for cognitive computing to determine the research needs that may exist [6].

A great deal of research has been done, and continues to be done, to determine the most effective means of making full use of each processor in a multiprocessor system for a wide variety of atomic problem types. In contrast, relatively little research has been done to determine effective means of managing overall application software development making use of these techniques. The rapidly growing cost of developing and maintaining application software for these systems is a symptom of this imbalance, and a cause for defense customers to be concerned. Computing systems will continue their growth into larger numbers of atomic units, with more varied interconnection topologies. Applications will continue to diversify in computational styles and increase in complexity. As these trends continue, the problems of managing software development and maintenance on platforms that are all but guaranteed to change during the course of a program will become a critical factor in the total cost of deploying high performance defense computing systems.

2.3   The Need for the Computer Cluster

GIT's contributions to the programs described above have been limited in the past by the lack of a dedicated computing resource to be used as a test bed for distributed multiprocessor software development frameworks. Each of these programs has need for a testing and reference implementation computing platform that is both general enough to be applicable to the wide variety of target platforms addressed, and also relevant to the multiprocessor configuration topologies likely to be faced in deployment. The Beowulf cluster purchased under this project will allow GIT to better support both current and future research in high performance embedded multiprocessor software.

Beowulf clusters have several unique characteristics that make them ideally suited to the research problems addressed by these programs, and provide high value compared to cost. The primary benefit is the ease with which Beowulf clusters can be composed of heterogeneous components. The relatively loosely-coupled topology, compared to other clustering systems, allows each node to achieve high efficiency largely independently of the specific nature of other nodes. The overall system is therefore not dependent on a homogeneous architecture to achieve high performance. This loose coupling also eases application deployment reconfiguration. A heterogeneous Beowulf cluster allows rapid redeployment of an application onto a wide variety of node topologies. This rapid reconfigurability can also be leveraged during the operation of an application to simulate resource reactive systems, test the software frameworks designed to use them, and test implementations of fault tolerant software systems.

Another benefit of this loose coupling is the low lifetime cost of purchasing, updating and maintaining the cluster, achieved through the modular coupling of a large amount of commodity general purpose computing hardware. The completely modular approach enables individual components to be removed, replaced, or upgraded independent of every other piece. It also enables easy expansion via addition of new computing resources. The resulting platform is scalable to future applications with only the cost of the incremental hardware. This modularity also greatly enhances the expected lifetime of the computing cluster. The cluster should be an effective test bed for multiprocessor computing for at least five to ten years, and with modular upgrades can be expected to be a useful test bed for up to fifteen years.

The new cluster allows GIT personnel and other users to perform more complete and realistic demonstration and testing of software systems for multiprocessor platforms. In particular, it is an extremely effective test bed for proposed products of the HPEC-SI program. The deployment platforms of interest in that program closely correspond to the topology of the cluster. Two main areas of interest in HPEC-SI are software frameworks that easily allow high utilization of parallel computing resources, and software frameworks that allow relatively rapid redeployment of applications onto different topologies of multiprocessor computers. The GIT cluster presents a very wide variety of potential topologies to applications, with the ability to reconfigure between them in a matter of seconds or less. This enables development of testing frameworks to evaluate reference implementations of the standards developed on the HPEC-SI program, and makes available a realistic development and testing platform to other participants in the HPEC-SI program via the Internet.

Similarly, the new system greatly enhances GIT's ability to expand its contributions to the PCA program. This program is centered on computing architectures that can be represented well by a rapidly reconfigurable cluster computer. Each of the processors being developed on the PCA program is divided internally into multiple sub-processors, each of which can usually be operated in one of several modes, with a reconfigurable communication interconnect network. The new GIT cluster is able to provide, at the macro level, an application deployment environment analogous in many ways to the abstract micro-architectures under development on the PCA program. The cluster allows simple and close administrative control of the number, configuration, and interconnect topology of processors available to an application, thus providing a computing platform that facilitates the development of a robust PCA simulation test bed. The cluster also supports the development of administrative resource management software, critical to the PCA program, which will control the available resources in response to test configuration, and the configuration of those resources according to PCA software solutions.

SECTION 3

THE CLUSTER COMPUTER SYSTEM

3.1 Cluster Design and Rationale

The deployed computing cluster is a heterogeneous system consisting of compute nodes based on 104 processors of varying number and architecture types, a single gigabit Ethernet interconnect communication network, a 2.5 terabyte network disk storage network, a control node, and associated rack enclosure hardware. The system contains 16 compute nodes that are each based around two AMD Opteron processors, 2 compute nodes that are based around four AMD Opteron processors, and 32 compute nodes that are based around two Intel Xeon processors. Each of the compute nodes is equipped with one gigabyte of local RAM storage per CPU, as well as approximately 35 gigabytes of local disk scratch storage. The theoretical peak performance of this cluster is over 900 gigaflops, giving a cost efficiency of over four peak theoretical megaflops per dollar, a very cost effective solution for modern computing hardware.

This variety of compute nodes allows testing software frameworks that are intended to deploy applications onto parallel computers of varying degrees of fine- and course-grained parallelism as well as varying network topologies. For example, since individual dual processor nodes can be rapidly rebooted into single-processor mode, the cluster presents three different node topologies to applications. An application configured for eight processors can be deployed on two four-processor nodes, four two-processor nodes, eight single processor nodes, or any combination thereof, with little or no downtime between runs. Since intra-node and inter-node parallelism exhibit differing behaviors, constraints, and tradeoffs, this redeployment capability represents an axis of comparison that is particularly relevant to current program development efforts. In addition, the most commonly used specialized network topologies (stars, rings, etc) are specializations of the general case all-to-all network configuration that trade restrictions on communication patterns for higher bandwidth or lower latency between network nodes. The number of compute nodes, as well as the varying number of processors used allows users to experiment with and compare communication organizations with the same constraints and trade-offs. The constraints of any network topology can be modeled by restricting the communication of compute nodes to other compute nodes. The degree of redundancy and variety among the compute nodes allows testing of deployment of applications onto subsets of the cluster that constitute a very wide array of target platforms. Additionally, these qualities allow testing of frameworks that model and simulate parallel platforms that are subject to hardware failures or reconfiguration during operation.

3.2 Equipment Purchased

The major system components were purchased from Penguin Computing, Inc. Additional minor components were purchased from GovConnection, Graybar, Inc., and Monarch Computer Systems, Inc. A detailed list of all equipment purchased, by vendor, is provided in the Appendix. The total cost was $185,526.46. The excess of $26.46 above the project budget of $185,500 was paid for out of Georgia state funds. Thus, all project funds were consumed in the purchase this equipment.

3.3 Installation

The cluster was installed at GTRI's Cobb County Research Facility (CCRF), located at 7220 Richardson Road, Smyrna, Georgia 30080, a suburb in the metropolitan Atlanta area. The system is housed in an access- and climate-controlled, raised-floor computing room. The system is connected to the Georgia Tech network and the external internet to provide access to both local and remote researchers. Figure 1 is a photograph of the cluster as installed at CCRF.
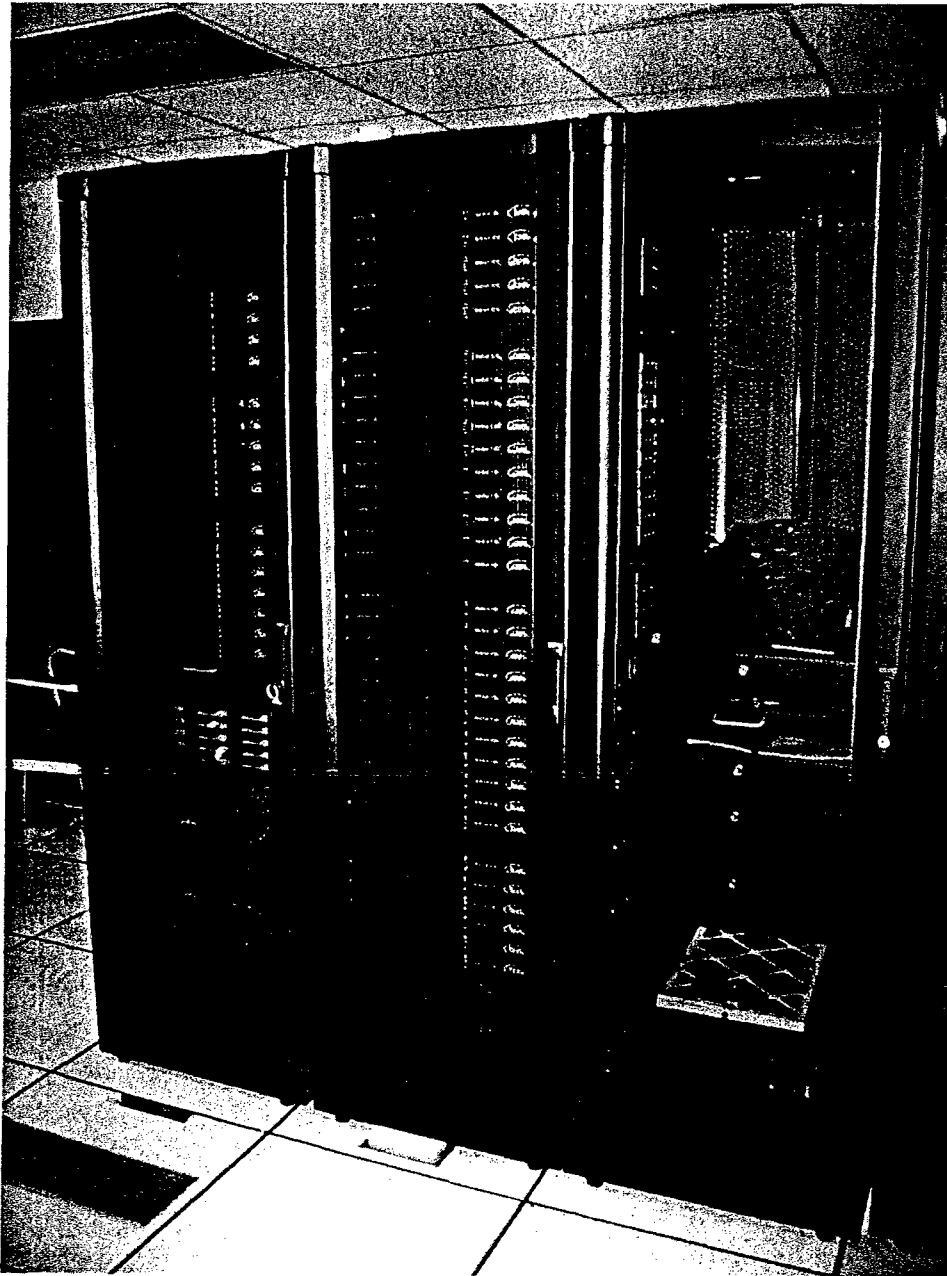

Figure 1. Photograph of the GIT cluster installed at the Cobb County Research Facility.

3.4 Current Operational Status

The deployed cluster system is currently fully operational, and is actively being used by users participating in several research programs. All compute nodes are operating properly. While several individual components had physical problems on delivery, all of these have been successfully addressed.

It had been determined prior to purchase that the cooling capacity of the machine room into which the cluster was installed was insufficient to handle the additional thermal load of the cluster. GTRI's Sensors and Electromagnetic Applications Laboratory upgraded the air conditioning capacity of the machine room so that it could host the system. No other facility modifications were required.

3.5 Open Issues

There are currently no open issues limiting the use of the system. It is recognized that there will be future costs associated with upkeep of the system. These include the costs of replacing out-of-warranty hardware as it fails, updating the control node system to support new software as it becomes available, and regular administrative support. It is expected that these costs will be borne through a combination of contractual support and support from GTRI overhead funds.

SECTION 4

INTERACTIONS AND TECHNOLOGY TRANSITIONS

4.1   Research Interactions

The deployed cluster is currently being used directly to support the HPEC-SI program [2]. Several participants of the HPEC-SI program, both within and outside GIT, are using the system to develop, test, evaluate, validate, and benchmark the development of implementations of parallel VSIPL++, as well as other systems for parallel software deployment. The system is also being used to test and evaluate the single-processor implementation of the VSIPL++ specification.

4.2   Training

The deployed cluster is supporting training in parallel software development for several users involved in the HPEC-SI program. GIT is actively supporting the development and testing the parallel VSIPL++ standard. Several of the users evaluating the parallel VSIPL++ specification and implementation have limited experience with parallel programming, and are receiving relevant training by use of the system.

In addition, a GIT student has been hired with the primary responsibility of providing system administration maintenance on the cluster. This student has relevant experience with system administration, but is currently using the cluster to augment training in administration of large parallel systems, as well as in the development of parallel software.

4.3   Publications

No publications specifically describing this system have been submitted or published, nor are any anticipated. Rather, it is expected that the cluster will be described primarily in future papers and reports focusing on software development techniques and experiments conducted using the cluster.

SECTION 5

FUTURE DEVELOPMENT

5.1 Additional Hardware

There are no current specific plans for additions to the cluster hardware. However, a number of possible growth directions exist, dependent on developments in high performance computing programs. The PCA program is actively examining extensions of the Morphware Stable Interface (MSI) to new hardware targets such as systems based on field-programmable gate arrays (FPGAs) and graphical processing units (GPUs). The addition of nodes using FPGA or GPU technology to the cluster would allow it to support research on systems using a mixture of heterogeneous, but fully-programmable microprocessors mixed with these specialized computing engines.

5.2 New Computational Programs

While there remains much to be done to develop effective programming paradigms for stream-based applications typical of DoD sensor processing, there is also a strong and increasing interest in cognitive processing and the computational requirements for such techniques [5],[6]. It is expected that the existing cluster may be well able to support experiments and software development in support of such "computational AI" (artificial intelligence) with its existing complement of node types.

REFERENCES

[1]   Vector, Signal, Image Processing Library (VSIPL) Forum web site, www.vsipl.org

[2]   High Performance Embedded Computing Software Initiative web site, www.hpec-si.org.

[3]   DARPA Polymorphous Computing Architectures (PCA) program web site,
        http://www.darpa.mil/ipto/programs/pca/index.htm.

[4]   Morphware Forum web site, www.morphware.org.

[5]   DARPA Architectures for Cognitive Information Processing (ACIP) program web site,
        www.darpa.mil/ipto/programs/acip/index.htm.

[6]   Cornell University and U.S. Air Force Research Laboratory, Research Directions in
        Architectures and Systems for Cognitive Processing workshop,
        www.csl.cornell.edu/wcas

APPENDIX

DETAILED LIST OF EQUIPMENT PURCHASES

The following tables provide detailed lists of equipment purchased form each vendor utilized.

Table 1. Components of the main system, purchased from Penguin Computing, Inc.

#	Part ID	Description	Qty
		Compute Nodes	
1	10001776	• Relion 130 – Compute Node • Dual 3.06GHz Intel P4 Xeon • 2GB Low Profile PC2100 ECC DDR (4 x 512MB) • 40GB, EIDE, 7200RPM • Penguin Remote Serial Management Card • Rackmount Ball-Bearing Rails (Rack Depth greater than 28") • Preload, ROCKS Version 3.2.0 Installation • Standard 3-Year Advanced Parts Replacement Warranty	32
2	10002304	• Relion 430 – Master Node • Hot-swap Power Supply; Dual 650 Watt Modules • Dual 3.06GHz Intel P4 Xeon • 6GB Low Profile PC2100 DDR (6 x 1GB) • Dual 3ware 7506-8 IDE Hardware RAID Controllers • 80GB, EIDE, 7200RPM (2MB cache) • RAID 5 Volume: 1436.2 GB (7+1×239.4 GB) • RAID 5 Volume: 1196.8 GB (6+1×239.4 GB) • Penguin Remote Serial Management Card • Slimline 8/24/10/24 IDE DVD-ROM/CD-RW • Intel Single Port Copper Gigabit Ethernet Card • Intel Dual Port 10/100 Ethernet Card • Preload, ROCKS Version 3.2.0 Installation • Standard 3-Year Advanced Parts Replacement Warranty	1
3	10002637	• Altus 4200 – Quad Opteron Node • Quad AMD Opteron 846 Processors • 4GB Low Profile PC2700 ECC DDR (4 x 1GB) • 36GB, 10,000RPM Low Profile SCA • CD/DVD-ROM Combo Drive • Penguin Remote Serial Management Card • Preload, ROCKS Version 3.2.0 Installation • Standard 3-Year Advanced Parts Replacement Warranty	2
4	10002429	• Altus 1000E – Dual Opteron Node • Dual AMD Opteron 244 Processors • 2GB Low Profile PC2700 ECC DDR (4 x 512MB) • 40GB, EIDE, 7200RPM • Penguin Remote Serial Management Card • Rackmount Ball-Bearing Rails (Rack Depth greater than 28") • Preload, ROCKS Version 3.2.0 Installation • Standard 3-Year Advanced Parts Replacement Warranty	16

Table 1 (continued). Components of the main system, purchased from Penguin Computing, Inc.

#	Part ID	Description	Qty
		Rack Enclosure and Peripherals	
5	10001731	NetShelter VX Base Enclosure 42UX600X1070mm AR2100 Black	1
6	10001750	Rack, AR2101BLK Expansion, 42U Black 1070mm	1
7	10002993	Packaging Wood Crate for APC Rack	2
8	10001187	Hardware Kit for Netshelter VX, 32 sets of cage nuts, bolts, and washers (AR8100)	4
9	10003184	Blanking Panel, 1U, Black, Hammond	7
10	10003185	Blanking Panel, 2U, Black, Hammond	2
11	10001025	Rack Mount LCD Monitor/Keyboard Drawer, Black (AR8215BLK)	1
12	10002953	UPS, APC Smart UPS 3000VA Rackmount 2U 208V USB/Serial (SUA3000RMT2U)	6
13	10003260	PDU Basic 1U 16A 208V (12)C13 APC AP9566	6
14	10003039	Cable, Power Cord, 250V .5 meter shielded	56
		Gigabit Network	
15	10001736	Switch, HP 4104gl Bare Switch, J4887A	1
16	10002933	Module, HP 20-Port GigE, J4908A	3
17	10002364	Cable CAT 6 10ft Yellow	20
18	10002365	Cable CAT 6 14ft Yellow	32
		Terminal Server Network	
19	10002828	Cyclades AlterPath ACS32 Advanced Console Server 32 port Single Power (ATP0100)	2
20	10001186	Cyclades Terminal Server Connection Cable (RJ45 to DB-9) (CAB0036)	52
		Total, including shipping and handling	**$177,685.00**

Table 2. Additional enclosure, Purchased from GovConnection.

#	Part ID	Description	Qty
1	AR2800BLK	• APC Netshelter VS 42U Enclosure Black, w/ sides	1
2	AR8215BLK	• APC 1u rackmount LCD Keyboard, Monitor, Mouse drawer	1
		Total, including shipping and handling	**$2,788.00**

Table 3. Additional accessories, Purchased from Graybar, Inc.

#	Part ID	Description	Qty
1	13082-X19	• Chatsworth Storage drawer for rackmount • 2U tall by 19 inches deep	1
2	13083-X19	• Chatsworth Storage drawer for rackmount • 3U tall by 19 inches deep	1
		Total, including shipping and handling	**$1,313.47**

Table 4. Additional control computer and network equipment, purchased from Monarch Computer Systems, Inc.

#	Part ID	Description	Qty
1	260202	ViewSonic VP201b Black 20.1" L	1
2	220205	Logitech Access Keyboard Enhan	1
3	230107	Logitech MX510 Mouse- RED	1
4	80302	Monarch Furia Custom Desktop	1
5	100029	THERMALTAKE Xaser III V1000A B	1
6	100365	Enermax 600W ATX EG701AX-VE-SF	1
7	110228	MSI K8N Neo2 Platinum Socket 9	1
8	120429	AMD Athlon 64 3500 90nm 939 P	1
9	800018	Thermal Grease, Shin-Etsu G675	1
10	140736	OCZ 1gb (2x512mb) EL DDR PC-32	1
11	150034	Seagate 7200.8 ST3300831AS 300	1
12	160947	NEC ND-3500A Dual Layer DVD±RW	1
13	170110	MITSUMI 3.5 FLOPPY DRIVE BLACK	1
14	190512	Connect3d Radeon X800 XT 256mb	1
15	210601	Power DVD XP 5.0 Software	1
16	800008	OPERATING SYSTEM(NONE) BARE B	1
17	800059	24/7 TECH SUPPORT+3 YR. DEPOT	1
18	140711	OCZ EL DDR PC-4400 / 550 MHz /	1
19	150137	WEST.DIGITAL WD2500JB,250GB,ID	1
20	280430	D-Link 5PORT 10/100/1000 Gigab	1
21	800011	Manufacturers warranty/no support	1
22	150561	Maxtor 6B300S0 MaXLine III 6B3	1
		Total, including shipping and handling	**$3,739.99**